

# 第41期-集成内存数据库访问

2020年11月16日 15:32

## ABP 与 内存数据库集成

Install-Package Volo.Abp.MemoryDb

```
[DependsOn(typeof(AbpMemoryDbModule))]
```

## 创建上下文

```
[ConnectionStringName("DogStore")]
public class DogMemoryDbContext : MemoryDbContext
{
    private static readonly Type[] EntityTypeList = {
        typeof(Dog)
    };

    public override IReadOnlyList<Type> GetEntityTypes()
    {
        return EntityTypeList;
    }
}
```

```
"ConnectionStrings": {
    "DogStore": "FE016679-1D65-4990-BAAE-3A1765B34F4A"
},
```

```
context.Services.AddMemoryDbContext<DogMemoryDbContext>(options =>
{
    options.AddDefaultRepositories<DogMemoryDbContext>(includeAllEntities:
true);
});
```

## 自定义扩展仓储

```
public class DogMemoryDbRepository : MemoryDbRepository<DogMemoryDbContext, Dog,
int>, IDogRepository
{
    public DogMemoryDbRepository(IMemoryDatabaseProvider<DogMemoryDbContext>
databaseProvider) : base(databaseProvider)
    {
    }

    public async Task DeleteDogsByAge(int age)
```

```

    {
        var entities = GetQueryable().Where(d => d.Age == age).ToList();
        foreach (var entity in entities)
        {
            await DeleteAsync(entity);
        }
    }
}

options.AddRepository<Dog, DogMemoryDbRepository>();

```

## 自定义内存数据库仓储基类

```

public class MyMemoryDbRepositoryBase<TEntity> :
    MemoryDbRepository<DogMemoryDbContext, TEntity> where TEntity : class, IEntity
    {
        public
        MyMemoryDbRepositoryBase(IMemoryDatabaseProvider<DogMemoryDbContext>
            databaseProvider) : base(databaseProvider)
        {
        }
    }

public class MyMemoryDbRepositoryBase<TEntity, TKey> :
    MemoryDbRepository<DogMemoryDbContext, TEntity, TKey> where TEntity : class,
    IEntity<TKey>
    {
        public
        MyMemoryDbRepositoryBase(IMemoryDatabaseProvider<DogMemoryDbContext>
            databaseProvider) : base(databaseProvider)
        {
        }
    }

options.SetDefaultRepositoryClasses(typeof(MyMemoryDbRepositoryBase<, >),
    typeof(MyMemoryDbRepositoryBase<>>));

```